

PR #45423 完整报告

vllm-project/vllm

[Bugfix] Correct prompt lengths for timed_traces benchmark

合并时间: 2026-08-14 05:51

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45423>

执行摘要

- 一句话: `timed_trace` 基准直接传 token ids, 修正 prompt 长度偏差
- 推荐动作: 值得精读, 尤其适合关注基准工具与 OpenAI API 协议兼容性的开发者。该 PR 展示了如何避免 tokenizer 非幂等导致的长度漂移, 以及通过类型注解与后端白名单双保险防止协议误用, 是一个小而完整的设计取舍案例。

功能与动机

PR body 明确指出 #39795 引入 `timed_traces` 后, 客户端按 trace 的 `input_length` (如 6758、7322) 构造请求, 但服务端 `tokenizer.encode(prompt)` 重新编码后实际收到的请求长度变成 7253、7844 等, 导致基准测得的 `request.num_tokens` 与 trace 预期不符, 破坏 `timed_trace` 的时序复现与性能对标。

实现拆解

1. 去掉 decode/encode 往返: 在 `vllm/benchmarks/datasets/datasets.py` 的 `TimedTraceDataset.sample()` 中删除 `prompt = tokenizer.decode(prompt_ids)`, `SampleRequest` 直接携带 `prompt_ids` (`list[int]`), `prompt_len` 仍取 `len(prompt_ids)`, 保证服务端按相同 token 序列编码时长度与 trace 声明一致。
2. 扩展类型注解: 将 `SampleRequest.prompt` 与 `RequestFuncInput.prompt` 的类型从 `str | list[str] | list[dict]` 扩展为 `str | list[str] | list[int] | list[dict]`, 使类型系统承认 token id 列表的合法传递路径 (对应 OpenAI completions 的 `prompt: list[int]` 支持)。
3. 后端兼容性校验: 在 `vllm/benchmarks/serve.py` 的 `main_async()` 中为 `timed_trace` 数据集添加 `args.backend not in ("vllm", "openai")` 的显式 `ValueError`, 阻止用户在 chat 类后端 (如 `openai-chat`) 中使用 `timed_trace` 时把 `list[int]` 静默当成字符串处理。
4. 测试与配套: 未新增测试文件, PR body 给出 `--max-concurrency 1` 的端到端手工验证步骤, 通过对比 `vllm/v1/core/sched/scheduler.py::Scheduler.schedule` 中的 `request.num_tokens` 确认修复效果。

关键文件:

- `vllm/benchmarks/datasets/datasets.py` (模块 基准工具; 类别 source; 类型 core-logic; 符号 `SampleRequest`, `TimedTraceDataset.sample`): 核心修复点: 移除 `decode` 步骤, `SampleRequest` 直接携带 token id 列表, 保证 prompt 长度与 trace 声明一致

- vllm/benchmarks/serve.py (模块 基准工具; 类别 source; 类型 core-logic; 符号 main_async) : 新增 timed_trace 数据集的后端白名单校验, 防止 chat 类后端静默处理 token 列表
- vllm/benchmarks/lib/endpoint_request_func.py (模块 基准工具; 类别 source; 类型 core-logic; 符号 RequestFuncInput) : 同步扩展 RequestFuncInput.prompt 类型注解, 让类型系统承认 token id 列表是合法输入

关键符号: main_async, TimedTraceDataset.sample

关键源码片段

vllm/benchmarks/datasets/datasets.py

核心修复点: 移除 decode 步骤, **SampleRequest** 直接携带 token id 列表, 保证 prompt 长度与 trace 声明一致

```
# vllm/benchmarks/datasets/datasets.py
# SampleRequest.prompt 的类型被扩展为接受 list[int],
# 允许直接把 token id 列表传给 OpenAI completions 接口的 prompt 字段
@dataclass
class SampleRequest:
    """
    Represents a single inference request for benchmarking.
    """

    prompt: str | list[str] | list[int] | list[dict]
    prompt_len: int
    expected_output_len: int = 0
    multi_modal_data: MultiModalDataDict | dict | list[dict] | None = None
    lora_request: LoRARequest | None = None
    request_id: str | None = None
    timestamp: float | None = None
    chat_messages: list[dict[str, Any]] | None = None
    request_overrides: dict | None = None

# TimedTraceDataset.sample() 中的关键改动:
# 之前会先 `prompt = tokenizer.decode(prompt_ids)` 再传字符串,
# 服务端 `tokenizer.encode(prompt)` 重新编码后长度会漂移 (decode 不幂等);
# 现在绕过 decode/encode, 直接把 token id 列表交给服务端,
# 服务端 `request.num_tokens` 与 trace 的 input_length 完全一致
prompt_ids = self._expand_prompt(
    entry.get(self.label_hash_ids, []), input_length, tokenizer
)
# 注意: 此处不再调用 tokenizer.decode()

prompt_len = len(prompt_ids)

samples.append(
```

```

SampleRequest(
    prompt=prompt_ids,
    prompt_len=prompt_len,
    expected_output_len=new_output_len,
    lora_request=None,
    multi_modal_data=None,
    request_id=request_id_prefix + str(ind),
    timestamp=timestamp,
)
)

```

vllm/benchmarks/serve.py

新增 `timed_trace` 数据集的后端白名单校验，防止 `chat` 类后端静默处理 `token` 列表

```

# vllm/benchmarks/serve.py main_async() 中 timed_trace 分支的变更:
# timed_trace 现在直接传递预 tokenized 的 prompt (list[int]) ,
# 只有 OpenAI completions 协议 (vllm 内部模拟或 openai 后端) 支持这种传法;
# chat 类后端会把 list[int] 当成字符串拼进消息体，静默产生错误请求，
# 所以这里显式拒绝其他后端，避免用户踩坑
if args.dataset_name == "timed_trace":
    if args.backend not in ("vllm", "openai"):
        raise ValueError(
            "timed_trace dataset passes pre-tokenized prompts (list[int])"
            " and requires a completions backend ('vllm' or 'openai')."
        )
    # timed_trace carries per-request timestamps;
    # ignore EOS so generation runs to the trace's specified output length,
    # and default to using those timestamps for scheduling unless the user
    # opted out.
    args.ignore_eos = True
    if args.self_timed is None:
        args.self_timed = True

```

评论区精华

review 中 `tdoublep` 对 `prompt=prompt_ids` 提出疑虑：`prompt_ids` 并不是真正的文本 `prompt`，是否在误用字段；`s3woz` 回应 OpenAI API 与 vLLM 的 `CompletionRequest.prompt` 官方支持 `list[int]` 直接传 `token id`，并顺带修正了 `endpoint_request_func.py` 的类型定义。随后 `tdoublep` 补充指出其他后端（如 `openai-chat`）不支持 `token` 传递，建议加后端校验，`s3woz` 认可并推送了校验代码，最终 `tdoublep` APPROVED。

- `prompt_ids` 是否误用 `prompt` 字段 (design): 确认 OpenAI completions 协议原生支持 `token id` 列表，属于合法用法，类型注解已更新
- 非 completions 后端的兼容性校验 (design): 在 `serve.py` 中为 `timed_trace` 添加 `args.backend` 白名单校验，非 `vllm/openai` 后端直接抛 `ValueError`

风险与影响

- 风险:

1. chat 后端兼容性风险: 直接传 `list[int]` 仅对 `completions` 后端有效, 若用户用 `openai-chat` 等后端运行 `timed_trace` 会得到静默错误; 本 PR 已通过后端白名单校验拦截, 但校验依赖 `args.backend` 字符串精确匹配, 后续若新增 `completions` 类后端需要同步更新白名单。
2. tokenizer 语义差异: `_expand_prompt` 生成的 `prompt_ids` 是否包含 BOS/ 特殊 token 取决于 `trace` 数据来源与 tokenizer 配置, 跳过 `encode` 后这些差异会原样传给服务端, 需保证 `trace` 数据本身与目标模型 tokenizer 兼容。
3. 缺少自动化测试: 本次改动没有对应单元测试文件, `prompt` 长度不匹配问题只能靠手工验证, 后续改动可能再次引入同类回归。 - 影响: 影响范围限定在 `vllm bench serve` 基准工具链: 修复 `timed_trace` 数据集的请求长度保真度, 使基于 `trace` 的时序基准 (`self-timed`、时间戳调度) 结果可信; 同时为 `SampleRequest/RequestFuncInput` 引入 `list[int]` 类型支持, 为未来其他预 tokenized 数据集复用该路径铺路。对生产推理服务无影响, 属于开发者工具侧的 bugfix。 - 风险标记: 缺少自动化测试, 依赖后端白名单字符串匹配

关联脉络

- PR #39795 [Feature] Add `timed_traces` support to `vllm bench serve`: 本 PR 修复的就是 #39795 引入的 `timed_trace` 功能中 `prompt` 长度失真的问题